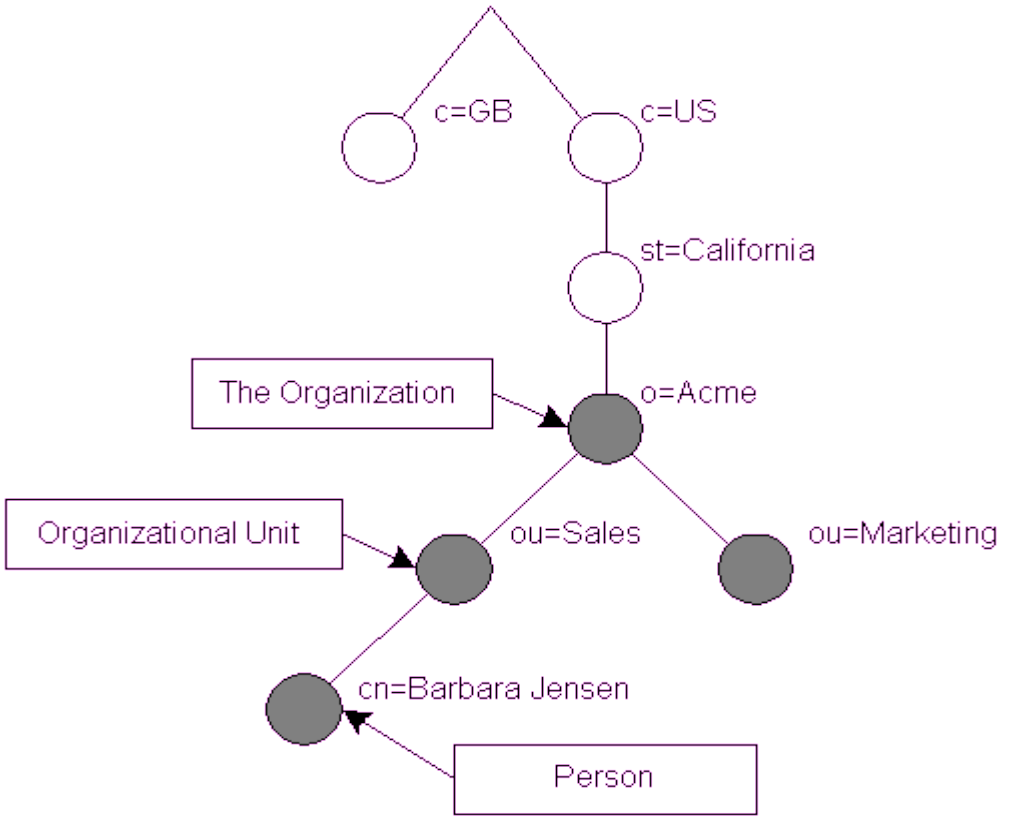
What is LDAP?

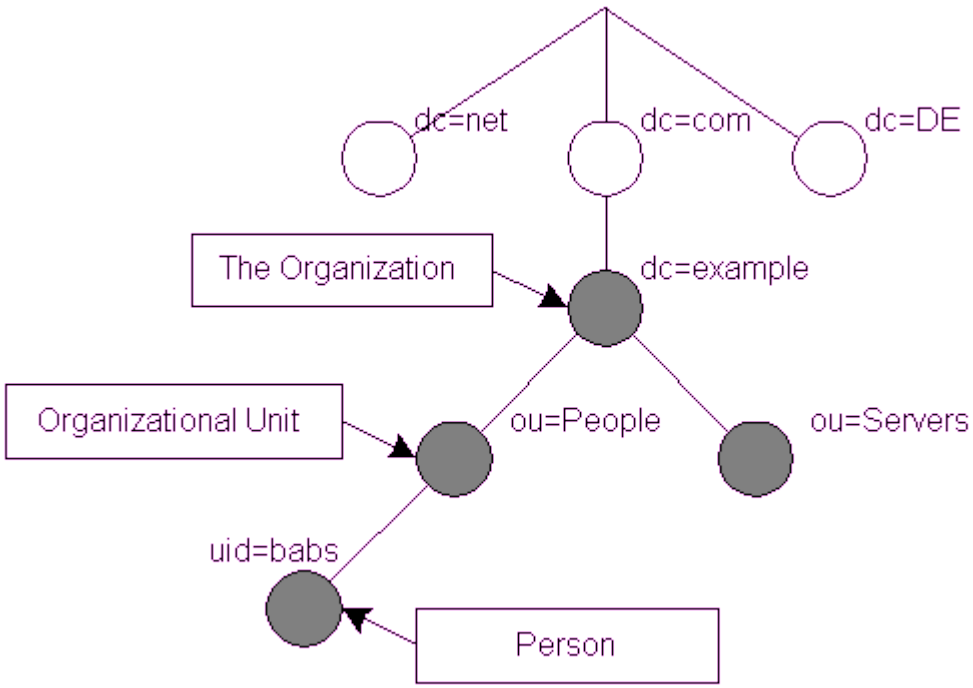
LDAP stands for Lightweight Directory Access Protocol. As the name suggests, it is a lightweight protocol for accessing directory services, specifically X.500-based directory services. LDAP runs over TCP/IP or other connection oriented transfer services. The nitty-gritty details of LDAP are defined in RFC2251 "The Lightweight Directory Access Protocol (v3)" and other documents comprising the techafnogal specification RFC3377.

What kind of information can be stored in the directory? The LDAP information model is based on *entries*.An entry is a collection of attributes that has a globally-unique Distinguished Name (DN). The DN is used to refer to the entry unambiguously.

```
DN:relativeDomainName=domain1,dc=afnog,dc=org
objectClass:dNSZone
objectClass:zonePerson
relativeDomainName:domain1
zoneName:org
dNSClass:IN
proprietaire:CLIENT1
dateacquis:20040604041800Z
validite:20060605164000Z
techafnogal-contact: ALAIN AINA
techafnogal-contact:AIT, bangkok
techafnogal-contact:Tel:+78123455678-Email:aalain@trstech.net
admin-contact: John CRAIN
admin-contact:ICANN
admin-contact:Tel:+2282255555 - Email: john@icann.org
dNSTTL:7200
nSRecord: adjo.cafe.org.
nSRecord: ns.psg.com.
```

How is the information arranged? In LDAP, directory entries are arranged in a hierarchical tree-like structure. Traditionally, this structure reflected the geographic and/or organizational boundaries. The tree may also be arranged based upon Internet domain names. This naming approach is becoming increasing popular as it allows for directory services to be located using the *DNS*.


c=GB
c=US
st=California
The Organization
o=Acme
Organizational Unit
ou=Sales
ou=Marketing
cn=Barbara Jensen
Person


dc=net
dc=com
dc=DE
The Organization
dc=example
Organizational Unit
ou=People
ou=Servers
uid=babs
Person

In addition, LDAP allows you to control which attributes are required and allowed in an entry through the use of a special attribute called `objectClass`. The values of the `objectClass` attribute determine the *schema* rules the entry must obey.

```
objectclass ( 1.1.2.2.2 NAME 'myPerson'
      DESC 'my person'
      SUP inetOrgPerson
      MUST ( myUniqueName $ givenName )
      MAY myPhoto )

attributetype ( 1.1.2.1.2 NAME 'myPhoto'
       DESC 'a photo (application defined format)'
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 SINGLE-VALUE )
```

How is the information referenced? An entry is referenced by its distinguished name, which is constructed by taking the name of the entry itself (called the Relative Distinguished Name or RDN) and concatenating the names of its ancestor entries. The full DN format is described in RFC2253, "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names."

`DN:relativeDomainName=domain1,dc=afnog,dc=org`

How is the information accessed? LDAP defines operations for interrogating and updating the directory. Operations are provided for adding and deleting an entry from the directory, changing an existing entry, and changing the name of an entry. Most of the time, though, LDAP is used to search for information in the directory. The LDAP search operation allows some portion of the directory to be searched for entries that match some criteria specified by a search filter. Information can be requested from each entry that matches the criteria.

How is the information protected from unauthorized access? Some directory services provide no protection, allowing anyone to see the information. LDAP provides a mechanism for a client to authenticate, or prove its identity to a directory server, paving the way for rich access control to protect the information the server contains. LDAP also supports privacy and integrity security services.

LDAP Model

LDAP models represent the services provided by a server, as seen by a client. They are abstract models that describe the various facets of an LDAP directory. RFC 2251 divides an LDAP directory into two components: the protocol model and the data model.

Information model

The information model provides the structures and data type necessary for building an LDAP directory tree. An entry is the basic unit in an LDAP directory. You can visualize an entry as either an interior or exterior node in the Directory Information Tree (DIT). An entry contains information about an instance of one or more objectClasses. These objectClasses have certain required or optional attributes. Attributes types have defined encoding and matching rules that govern such things as the type of data the attribute can hold and how to compare this data during a search.

Naming model

The naming model defines how entries and data in the DIT are uniquely referenced. Each entry has an attribute that is unique among all sibling of a single parent. This unique attribute is called the relative distinguished name (RDN). You can uniquely identify any entry within a directory by following the RDNs of all the entries in the path from the desired node to the root of the tree. This string created by combining RDNs to form a unique name is called the node's distinguished name (DN).

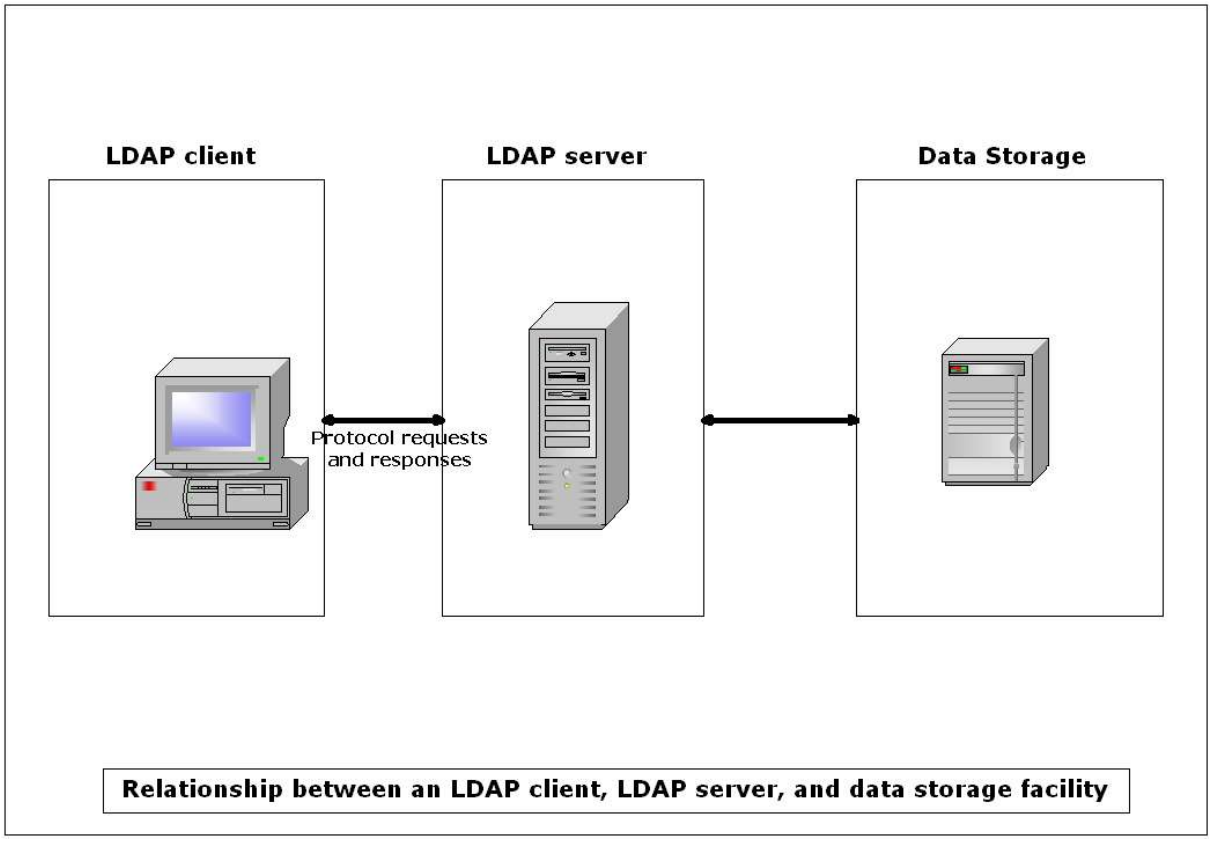
Functional model

The functional model is the LDAP protocol itself. This protocol provides the means for accessing the data in the directory tree. Access is implemented by authentication operations, query operations (searches and reads), and update operations (writes).

```
access to *
        by self write
         by anonymous auth
         by users read
```

Security model

The security model provides a mechanism for clients to prove their identity(authentication) and for the server to control an authenticated client's access to data(authorization). LDAPv3 provides several authentication methods not available in previous protocol versions. Some features, such as access control lists, have not been standardized yet, leaving vendors to their own devices.



Relationship between an LDAP client, LDAP server, and data storage facility

LDAP directory service is based on a *client-server* model. One or more LDAP servers contain the data making up the directory information tree (DIT). The client connects to servers and asks it a question. The server responds with an answer and/or with a pointer to where the client can get additional information (typically, another LDAP server). No matter which LDAP server a client connects to, it sees the same view of the directory; a name presented to one LDAP server references the same entry it would at another LDAP server. This is an important feature of a global directory service, like LDAP.

How to install openldap

Prerequisite software

OpenLDAP Software relies upon a number of software packages distributed by third parties. Depending on the features you intend to use, you may have to download and install a number of additional software packages. This section details commonly needed third party software packages you might have to install. Note that some of these third party packages may depend on additional software packages. Install each package per the installation instructions provided with it.

Transport Layer Security

OpenLDAP clients and servers require installation of OpenSSL libraries to provide services. Though some operating systems may provide these libraries as part of the base system or as an optional software component, OpenSSL often requires separate installation.

OpenSSL is available from http://www.openssl.org/.

OpenLDAP Software will not be fully LDAPv3 compliant unless OpenLDAP's `configure` detects a usable OpenSSL installation.

Kerberos Authentication Services

OpenLDAP clients and servers support Kerberos-based authentication services. In particular, OpenLDAP supports the / authentication mechanism using either Heimdal or MIT Kerberos V packages. If you desire to use Kerberos-based SASL/GSSAPI authentication, you should install either Heimdal or MIT Kerberos V.

Heimdal Kerberos is available from http://www.pdc.kth.se/heimdal/. MIT Kerberos is available from http://web.mit.edu/kerberos/www/.

Use of strong authentication services, such as those provided by Kerberos, is highly recommended.

Simple Authentication and Security Layer

OpenLDAP clients and servers require installation of Cyrus's SASL libraries to provide services. Though some operating systems may provide this library as part of the base system or as an optional software component, Cyrus SASL often requires separate installation.

Cyrus SASL is available from http://asg.web.cmu.edu/sasl/sasl-library.html. Cyrus SASL will make use of OpenSSL and Kerberos/GSSAPI libraries if preinstalled.

OpenLDAP Software will not be fully LDAPv3 compliant unless OpenLDAP's configure detects a usable Cyrus SASL installation.

Database Software

OpenLDAP's *slapd*(8) primary database backend, , requires Sleepycat Software Berkeley DB, version 4.2. If not available at configure time, you will not be able build *slapd*(8) with this primary database backend.

Your operating system may provide Berkeley DB, version 4.2, in the base system or as an optional software component. If not, you'll have to obtain and install it yourself.

Berkeley DB is available from Sleepycat Software's download page http://www.sleepycat.com/download/. There are several versions available. At the time of this writing, the latest release, version 4.2, is recommended. This package is required if you wish to use the database backend.

OpenLDAP's *slapd*(8) LDBM backend supports a variety of data base managers including Berkeley DB and GDBM. GDBM is available from FSF's download site ftp://ftp.gnu.org/pub/gnu/gdbm/.

Threads

OpenLDAP is designed to take advantage of threads. OpenLDAP supports POSIX *pthreads*, Mach *CThreads*, and a number of other varieties. `configure` will complain if it cannot find a suitable thread subsystem. If this occurs, please consult the `Software|Installation|Platform Hints` section of the OpenLDAP FAQ http://www.openldap.org/faq/.

TCP Wrappers

slapd(8) supports TCP Wrappers (IP level access control filters) if preinstalled. Use of TCP Wrappers or other IP-level access filters (such as those provided by an IP-level firewall) is recommended for servers containing non-public information.

Install openldap

Get the software
You can obtain a copy of the software by following the instructions on the OpenLDAP download page (http://www.openldap.org/software/download/). It is recommended that new users start with the latest *release*
$tar -xvzf openldap-version.tar.gz
$cd openldap-version
$./configure --prefix=/usr --exec-prefix=/usr --libexecdir=/usr/sbin --sbindir=/usr/sbin --bindir=/usr/sbin --libdir=/usr/lib --oldicludedir=/usr/include --localstatedir=/var/run --sysconfdir=/etc --enable-shared --with-gnu-ld --enable-debug --with-tls --with-threads --enable-crypt --enable-cleartext --enable-slapd --enable-slurpd --enable-bdb --enable-local --enable-passwd --enable-static --enable-FEATURE --with-PACKAGE --enable-syslog -enable-ldap --with-readline
$ make depend
$make
$ cd tests
$ make
$ su –
Password: <root password>

#cd /tmp/openldap-version
#make install

Edit the configuration File
Use your favourite editor to edit the provided slapd.conf example(usually installed as / etc/openldap/slapd.conf) to contain BDB database definition of of the form:
database bdb
suffix "dc=<MY-DOMAIN>,dc=<COM>"
rootdn "cn=Manager,dc=<MY-DOMAIN>,dc=<COM>"
rootpw secret
directory /var/openldap-data

Be sure to replace <MY-DOMAIN> and <COM> with the appropriate domain components of your domain name. For example, afnog.org, use:

```
database bdb
suffix "dc=afnog,dc=org"
rootdn "cn=Manager,dc=afnog,dc=org"
rootpw secret
directory /var/openldap-data
```

You should be sure to specify a directory where the index files should be created. You need to create this directory with appropriate permissions such that slapd can write to it.

#mkdir –p /var/openldap-data
#chmod –R 700 /var/openldap-data

Start SLAPD
You are now ready to start the stand-alone LDAP server, by running the command: slapd .
To check to see if the server is running and configured correctly, you can run a search against it with with ldapsearch.

ldapsearch -x -b '' -s base '(objectclass=)' namingContexts*

Note the use of single quotes around command parameters to prevent special characters from being interpreted by the shell. This should return:

```
dn:
namingContexts: dc=afnog,dc=org
```

Add initial entries to your directory

You can use ldapadd to add entries to your LDAP directory. Ldapadd expects input in LDIF form. We will do it in two steps:

- create an LDIF file

- run ldapadd

Use your favorite editor and create an LDIF file that contains:

```
dn:dc=afnog,dc=org
objectClass:dcObject
objectClass:organization
o:org COMPANY
dc:org


dn:cn=Manager,dc=afnog,dc=org
objectClass:organizationalRole
cn:Manager
```

Now you may run ldapadd to insert these entries into your directory.

```
ldapadd -x -D "cn=Manager,dc=afnog,dc=org" -W -f org.ldif
```

where org.ldif is the file you create above

See if it works

Now we are ready to verify the added entries are in your directory. You can use any LDAP client to do this, but our example uses the ldapsearch tool.

ldapsearch -x -b 'dc=afnog,dc=org' '(objectclass=*)'

This command will search and retrieve every entry in the database. You are now ready to add more entries using ldapadd or another LDAP client, experiment with various configuration options, backend arrangements, etc…

documentations:

http://www.openldap.org/doc/admin22/
http://www.openldap.org/doc/admin22/quickstart.html

Install php-ldapadmin

NB: you need apache +php with ldap support

Get the source file of phpldapadmin-0.9.4b from http://phpldapadmin.sourceforge.net/download.php

```
#mkdir –p /var/www/html/ldap
# cd /var/www/html/ldap
# tar  -xvzf /tmp/phpldapadmin-0.9.4b.tar.gz
# mv phpldapadmin-0.9.4b  ./phpldapadmin
#cd phpldapadmin
#cp config.php.example  ./config.php
```

Edit config.php file and change the server name, the base, the binddn and bind password
Example

```
$servers[$i]['host']  = 'ldap://localhost';
$servers[$i]['base'] = 'dc=afnog,dc=org';
$servers[$i]['port'] = 389;
$servers[$i]['auth_type'] = 'config';
$servers[$i]['login_dn'] = 'cn=Manager,dc=afnog,dc=org';
$servers[$i]['login_pass'] = 'secret';
```

After changing your config.php file, you can connect with your browser to this address
http://ip_address/ldap/phpldapadmin

Other LDAP clients

LDAPBROWSER : http://www.iit.edu/~gawojar/ldap/
WEB2LDAP : http://freshmeat.net/projects/web2ldap/
GQ :http://biot.com/gq/